Core CCSLM

A simple primitive with non-trivial consequences

The core of a Concatenated Context-Selected Linear Mapping (CCSLM) is remarkably simple:

1. Context selects a linear mapping.
2. The selected mapping is applied.
3. The result is passed to the next computational stage.
4. The process is repeated and the complete system is trained by backpropagation.

For a sequence of stages,

$$x_{i+1} = L_i(C_i)x_i$$

and therefore

$$y = L_n(C_n) \cdots L_2(C_2)L_1(C_1)x.$$

For any fixed sequence of contexts, the complete CCSLM is mathematically equivalent to a single linear mapping

$$y = Mx, \qquad M = L_n \cdots L_1.$$

This apparent simplicity is deceptive.

Matrix concatenation creates a new effective mapping

The product of two matrices is generally not simply "another version" of either matrix. If

$$M = BA,$$

then M can have substantially different rank, singular-value structure, null space, conditioning, noise sensitivity and associative-memory behaviour from both A and B.

Thus a chain of simple linear mappings continually creates new effective mappings.

This produces an important CCSLM distinction:

> **The individual mappings are simple, but their composition can be a substantially different computational object.**

Telescoping associative memory

The same matrix product can be interpreted as a hierarchy of associative recalls.

Consider

$$x \xrightarrow{A} h_1 \xrightarrow{B} h_2 \xrightarrow{C} y.$$

The system performs three successive linear associative mappings:

$$h_1 = Ax, \qquad h_2 = Bh_1, \qquad y = Ch_2.$$

Yet the intermediate recalls can be telescoped exactly:

$$y = CBAx = Mx.$$

From the outside, M is just one linear associative mapping. Internally, however, it represents the telescoped effect of several successive recalls.

This gives two complementary views:

- **Sliding view:** each matrix performs an association between adjacent representational spaces.
- **Magnifying view:** the complete product is one association between the original input and final output.

The effective matrix can therefore be regarded as a **telescoped associative memory containing the effect of multiple intermediate recalls**.

This is particularly relevant to the view of deep linear systems as hierarchical associative memories.

Backpropagation trains the factorization, not merely the effective matrix

Although

$$M = W_n \cdots W_1$$

is equivalent to a single matrix for forward computation, training the factors $W_1, \ldots, W_n$ is not equivalent to directly training M.

Backpropagation modifies the individual factors. Consequently, the *factorization* has learning dynamics that are invisible if one only examines the final effective matrix.

This connects directly with the work of Andrew Saxe and collaborators on deep linear networks. Their analysis demonstrated that deep linear networks, despite being equivalent to a single linear transformation in the forward direction, possess non-trivial gradient-learning dynamics, including structured learning of different modes of the transformation.

Thus:

> **Forward equivalence does not imply learning equivalence.**

The effective matrix describes what the trained system does; the factorization describes how that behaviour is constructed and learned.

Context-selected factorization

CCSLM adds context selection to this picture:

$$M(C) = L_n(C_n) \cdots L_1(C_1).$$

Different context sequences therefore produce different effective matrices. The system can be viewed as learning an **atlas of effective linear mappings**, each generated by a sequence of selected regional linear experts.

The regional mappings are individually simple, but their compositions can be very different.

Best-average linear mappings

Backpropagation also gives a useful interpretation of the individual regional mappings.

A mapping only receives training signals from the data routed to it by its context. Consequently, each selected mapping is driven toward a useful, approximately **best-average linear mapping for the slice of the data distribution that it sees**.

This provides a simple interpretation of context specialization:

- If two context states consistently expose a mapping to similar data, their learned mappings may remain similar.
- If they expose substantially different data, their mappings can specialize.
- Several context states may form clusters that share similar mappings.

This motivates the **Context Specialization Metric (CSM)**: measure how different the learned mappings become as a function of context.

Rank, null space and observability

The telescoped matrix also exposes an important limitation.

For

$$M = L_n \cdots L_1,$$

the composition can develop a large null space or otherwise lose rank. Information discarded at an intermediate stage cannot generally be recovered by later linear mappings.

Consequently, **observability of the original input through the data pathway is a fundamental property of a CCSLM composition**.

However, CCSLM has a second pathway: context.

Information that is no longer observable in the transformed data may still be represented by the context used to select subsequent mappings. Thus context can potentially restore system-level observability even when the data pathway itself is rank deficient.

This suggests treating data and context as distinct information channels:

> **The data pathway carries quantities being transformed; the context pathway can carry information about the state in which that transformation should occur.**

A simple transpose view of backpropagation

For a linear mapping

$$y = Lx,$$

the backward propagation of an error signal uses the transpose:

$$\delta x = L^T \delta y.$$

In a context-selected system, only the mappings selected by the current context participate in the forward computation and receive the corresponding backward update. Non-selected mappings receive zero gradient for that example.

This makes CCSLM backpropagation particularly easy to conceptualize:

> **Context selects the forward mappings; the backward signal follows the transpose of the selected mappings.**

The derivative machinery remains useful mathematically, but is not essential to the conceptual description.

The core idea

The core CCSLM mechanism can therefore be stated in one sentence:

> **Context selects linear mappings, selected mappings are concatenated into an effective linear transformation, and backpropagation learns the factorized mappings from the data slices routed to them.**

Despite the simplicity of the primitive, this produces several distinct objects of study:

- the **regional linear mappings**;
- their **telescoped effective matrices**;
- the **associative-memory hierarchy** represented by their factorization;
- the **learning dynamics** of that factorization;
- the **rank, null-space and observability** of the resulting compositions;
- the **information carried by the context pathway**;
- and the **degree of context specialization** of the learned mappings.

The computational primitive is small. The mathematical structure generated by repeatedly applying it is not.